

APPLICATION  
FOR  
UNITED STATES LETTERS PATENT

TITLE: NAVIGATION IN RENDERED THREE-DIMENSIONAL SPACES

APPLICANT: JOHN J. LIGHT, JOHN D. MILLER, AND DOUG L. SORENSON

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL584937194US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

Date of Deposit May 21, 2001

Signature 

Gil Vargas  
Typed or Printed Name of Person Signing Certificate

Application for United States Patent

in the name of John J. Light (Beaverton OR), John D. Miller  
(Portland OR), and Doug L. Sorenson (Hillsboro OR) for

NAVIGATION IN RENDERED THREE-DIMENSIONAL SPACES

BACKGROUND

This invention relates to navigation in rendered three-dimensional (3D) spaces.

A 3D space can be displayed, for example, as a 2D rendering on a flat surface of a monitor or as a pair of stereo images, which can be viewed by a trained operator using stereo-glasses or a stereo-projection headpiece. Displayed 3D spaces can be used for simulations, such as flight simulators and fantasy games, design, and information visualization.

A displayed 3D space can provide an operating environment in which files, information, and applications are represented as objects located in the space. The WebBook and Web Forager environments used 3D space to organize and retrieve web pages (Card et al. "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web," in Proceedings of CHI'96 (New York, NY) 1996 ACM Press 111-117). The STARLIGHT Information Visualization System provided an integrative information display environment in which the user's viewpoint could navigate in a 3D coordinate space (Risch et al. "The

STARLIGHT Information Visualization System," in *Proceedings of*  
IV '97 (London UK, August 1997) IEEE Computer Society 42-49).

The Task Gallery is a 3D environment for document handling and  
task management (see, e.g., Robertson et al. "The Task

5 Gallery: A 3D Window Manager," in *Proceedings of CHI'2000*,"  
(The Hague NL, April 2000), ACM Press, 494-501).

The navigation of 3D space is facilitated by locating the  
3D position of a user's interest using controls originally  
designed for navigation of 2D space. U.S. Patent Nos.

10 5,689,628, and 5,608,850 describe methods of coupling a user's  
viewpoint in the 3D space to the transport of objects in the  
3D space.

#### DESCRIPTION OF DRAWINGS

15 Figures 1A and 1B are schematics of a system for  
operating Miramar, a simulated 3D environment for handling  
files and objects.

Figures 2A and 2B are a line drawing and a screenshot of  
a 2D projection of a 3D space.

Figures 3A, 3B and 3C are schematics of a 3D space.

20 Figure 4 is a flow chart of a process for tracking a  
center of interest (COI).

Figure 5 is a diagram of available directions of movement  
relative to a COI.

25 Figure 6 is a flow chart of a method of selecting an  
object.

**DETAILED DESCRIPTION**

The so-called Miramar program is one implementation of aspects of the invention. Miramar simulates a 3D environment for file and object management. Referring to Figure 1A, 5 Miramar runs on a computer 110 that is interfaced with a monitor 120, a keyboard 130, and a mouse 140. As shown in figure 1B, the computer 110 can include a chipset 111 and central processing unit (CPU) 112 that operates Microsoft Windows® and that can compute 2D screen renderings of 3D 10 space. The chipset 111 is connected to system memory 113. The computer 110 includes I/O interfaces 115, 116, and 118 for receiving user controls from the keyboard 130 and mouse 140.

The computer 110 also includes an interface 114 for video output to the monitor 120. Referring also to Figures 2A and 15 2B?, the Miramar program generates a window 180 that is rendered on a 2D display area 125 of the monitor 120.

Referring also to the examples in Figures 3 and 4, the program displays 410 a first 2D projection 305 of a 3D space 310 to a user 10. The space 310 can include an object 330 20 that is located at a particular 3D location, and, in the example in Figure 3A, is not visible in the first projection 305. The projection 305 is relative to a first point of reference (POR) 320. Information about the location of objects 330 in the 3D space 310 and the current POR 320 can be 25 stored in the system memory 113.

Referring to the examples in Figures 2A and 2B, the projection of the 3D space 310 includes a planar surface 200,

topographical elements 210, and objects 220, and the display also shows an indicator 250, and a cursor 290.

The topographical elements 210 can be selected by the user from a variety of scenes, such as mountains, fjords, canyons, and so forth. The topographical elements 210 provide a sense of scale and depth.

The planar surface, or "floor" 200 is rendered as a finite square grid with grid lines 205 and 206. For example, the grid lines 206 that project into the scene 180 are angled in perspective to meet at a vanishing point 214 on the horizon 212. The grid lines 205 and 206 enhance the user 10's sense of perspective. When projected, the floor 200 is generally oblique to the display area 180, except, of course, when the POR 320 is directly overhead.

The planar surface 200 can include landmarks such as a cone 280 that is positioned at its center. The cone provides a reference point for the user 10, called "home."

The planar surface 200 features an indicator 250, which can be a squat cylinder or "puck," for example, as depicted in Figure 2B.

Referring also to Figure 5, the indicator 250 provides a reference for the user 10 of the center of interest (COI) 560. The COI 560 is typically above the surface 200, and the indicator 250 is constrained to the surface 200 so as not to obscure the display of objects 220 in the scene 305. The user 10 can also control the indicator 250 as described below.

The 3D space 310 also includes objects 220, such as bulletin boards 222, notes 224, web pages 226, and text 228 that are rendered in positions above the surface 200. A "shadow" 260 of each object 220 is displayed on the surface 200 at a position that is directly underneath the object 220, such that a line between the shadow 260 and the object 220 is normal to the surface 200 in the 3D space 310. The shadows 260 orient the user 10 when navigating on the surface 200.

The user 10 can rely on visual recognition of the objects 220, topographic features 210, shadows 260, and gridlines 205 and 206 to orient himself in the coordinate space 310 and infer his point of reference 320.

At least five modes can be used to navigate in Miramar. Generally, navigation is controlled by the keyboard 130 and mouse 140. In some of the modes, the user can interface with at least two indicators, one being the indicator 250, the other being the cursor 290.

The first mode of operation enables the user 10 to reorient with respect to a COI 560, typically without moving the user's point of reference 320.

Referring to the example in Figures 3A and 3B, the program displays a first view 305 of the 3D space 310. The program allows the user 10 to select the indicator 250, e.g., using the cursor 290, which is controlled by the mouse 140. The selection of the indicator 250 is detected 420 and subsequently user controls (e.g., of the mouse 140) are coupled 430 to movement of the indicator 250 along the surface

200. For example, user-directed movement of the mouse 140 along each of the two axes on a table is translated into scaled movement of the indicator 250 on the 2D plane 200.

When the program detects 440 an event such as release of a mouse button, the program alters the window 180 to display a second view 340 based on the new position of the indicator 250. Other events that can be detected include: an arrest of movement of the indicator 250; or movement of the indicator 250 to a margin of the first view 305 or outside the first view 305. The latter event can be used to enable the user 10 to pan through the space 310.

The alteration to the rendering of window 180 can be a rotation about the POR 320, i.e., the location of the user's position in 3D space 310 is the same, but the angle of the user's view of the 3D coordinate space 310 is altered from the first view 305 to a second view 340. Typically, the second view 340 locates the COI 560 in the center of the 2D display area 180. The level of the horizon 212 can also be adjusted so that the COI 560 is visible.

The alteration of the window 180 from the original view 305 to the second view 340 can be rendered in a seamless manner. For example, the program may display a sequence of views with respect to time that simulate to the user 10 a rotation and/or tilting of his head with respect to the space 310.

In a second mode of navigation, the user 10 moves his POR 320 in any of three dimensions with respect to the COI 560, as

depicted in Figure 5. The user 10 uses the cursor 290 coupled to the mouse 140 to navigate. The cursor is used to select directional buttons on the control panel 270. Keyboard 130 strokes (e.g., of arrow keys) also function to receive user moves.

Left and right commands rotate the user's POR 320 in a circular orbit 530 around the COI 560. The POR 320 is moved at a constant angular velocity about the axis 550 at the COI 560. The angular velocity used is independent of distance from the indicator 250. The circular trajectory around the COI 560 allows the user 10 to see all facets of an object at the COI 560.

Up and down commands can be used to increase and decrease the inclination 540 of the user's POR 320 with the respect to the COI 560. Movements in this direction can also be made in an orbital path 540 with a constant angular velocity.

Zoom in and out commands can be used to alter the distance 520 between the user's POR 320 and the COI 560. These movements can be made with an effective velocity that is proportional to the distance. Typically, a standard increment, e.g., for a keyboard command for zoom movements, is a distance that is approximately 6% of the distance from the current viewpoint to the COI 560. Such scaling prevents the user 10 from advancing past the COI 560.

In a third mode of navigation, the user 10 manipulates 430 the indicator 250 to specify a COI 560. Then in response to an event 440, the program displays a second view 360 from a



second POR 350, as illustrated in Figure 3C. For example, the event can be release or double-clicking of a mouse button.

The second view 360 can include an alteration that enhances the representation of the COI 560. For example, the second position 350 can provide a second view 360 that enlarges the COI 560 and/or provides a view of a primary facet of the COI 560.

The program can again provide an apparently seamless transition from the first view 305 or 340 to the second view 360 by displaying a sequence of views, such that the user perceives he is flying on a trajectory 355 through the 3D space 310 from the original position 320 to the second position 350.

In a fourth mode of navigation, the user 10 again manipulates 430 the indicator 250 to specify a COI 560. In response to an event 440, such as a double mouse click, the program identifies an object 330 based on the position of the indicator 250. Typically, the identified object 330 is the object that is located directly above the indicator 250.

Otherwise, the object that is closest to the indicator 250 can be used. The program then triggers 470 a process that is associated with the selected object 330.

In Miramar, many objects represent links to files. The triggered process can include activating an application appropriate for the linked file to open or read the linked file. Other objects can represent web links, which when

selected open up the corresponding web site using the default web browser.

The use of the indicator 250 to specify an object is particularly cogent when objects 220 are partially or completely overlapping in a particular rendering of the 3D space.

In a fifth mode of navigation, the user 10 selects an object or point of interest in the 3D space 310 using the cursor 290. The program identifies the coordinates of the cursor 290 position, and then determines if an object 330 is displayed at that position in the current rendering of the 3D space 310. If an object is present, it is designated the selected object 330. Otherwise, the position is designated as a selected point. In addition, the user can select an object of interest using a text menu that lists available objects by their identifiers.

After an object or point is selected, the indicator 250 is repositioned automatically underneath the selected object 330 or point to confirm to the user the new COI 560 defined by the selection event. If no object is present or visible at the selected point, the indicator 250 can serve as a surrogate for an object to the user 10.

The program includes other optional features that can be activated to assist the user in selecting objects 220 with the indicator 250. For example, the indicator 250 can be rendered with a projection that extends normal to the surface 200 to the height of an object located above the indicator 250. In

still other implementations, an object located above the indicator is rendered differently, e.g., highlighted with a color or assigned a new attribute (e.g., "flashing," and so forth).

5 Other implementations are also within the scope of the claims. For example, although the Miramar program provides a 3D space for managing files and information, the featured indicator 250 can be used in any program that renders a projection of 3D space. Other programs can include computer-  
10 assisted design applications, defense and security applications, cartographic applications, mathematical modeling applications, games, and simulators.

15 In some implementations, two surfaces 200 are used that are normal to each other. One surface is located in the x-y plane, whereas the other is in the y-z plane. Each surface has an indicator 250 linked to the position of a COI such that a line between each indicator and the COI is normal to its respective surface. Thus, the user 10 can readily perceive the position of the COI 560 in 3D space 310 as rendered in a  
20 2D projection.

In other implementations, the 2D surface 200 is not planar, e.g., it is concave or convex. Positions on the 2D surface are nevertheless addressable using two coordinates, e.g., Cartesian or non-Cartesian coordinates.

The monitor 120, mouse 140, and keyboard 130 can be replaced by other user interfaces such as stereo headpieces, joysticks, and so forth.

The techniques described here are not limited to any particular hardware or software configuration; they may find applicability in any computing or processing environment. The techniques may be implemented in hardware, software, or a combination of the two. The techniques may be implemented in programs executing on programmable machines such as mobile or stationary computers, personal digital assistants, and similar devices that each include a processor, a storage medium readable by the processor, at least one input device, and a display.

Each program may be implemented in a high-level procedural or object oriented programming language to communicate with a machine system. However, the programs can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language.

Each such program may be stored on a storage medium or device, e.g., compact disc read only memory (CD-ROM), hard disk, magnetic diskette, or similar medium or device, that is readable by a general or special purpose programmable machine for configuring and operating the machine when the storage medium or device is read by the computer to perform the procedures described in this document. The system may also be implemented as a machine-readable storage medium, configured

with a program, where the storage medium so configured causes  
a machine to operate in a specific and predefined manner.

03222350-03222350